

CLAIMS

I/We claim:

[c1] 1. A method in a computer system for creating a file system snapshot, the data of the file system being organized hierarchically via nodes, the method comprising:

copying a root node of the file system to a new node that points to the same child nodes of the root node, the new node represents a root node of the snapshot; and

when a node of the file system is modified,

replacing ancestor nodes of the node that have not yet been replaced with a new node;

replacing the node with a new node that points to the same child nodes of the replaced node; and

effecting the modification on the new node.

[c2] 2. The method of claim 1 wherein when multiple snapshots occur, the ancestor nodes of the node to be modified that are replaced are those ancestor nodes that have not yet been replaced during the current snapshot.

[c3] 3. The method of claim 2 wherein each new node has a snapshot identifier that identifies the snapshot during which it replaced a node and including checking the snapshot identifier of an ancestor node to determine whether it has been replaced during the current snapshot.

[c4] 4. The method of claim 3 wherein when a node is not to be part of a snapshot, associating an indication with that node so that node will not be replaced when it or any descendent node is modified.

[c5] 5. The method of claim 1 wherein when the snapshot is accessed via the root node of the snapshot.

[c6] 6. The method of claim 1 wherein each new node has an identifier that is different from the identifier of the node it replaced.

[c7] 7. The method of claim 6 including associating the identifier of the replacing node with the replaced node so that, when a request to access a node identified by the identifier of the replaced node is received, that association is used to access the replacing node.

[c8] 8. The method of claim 7 wherein the associating includes storing the identifier of the new node in the replaced node.

[c9] 9. The method of claim 7 including associating the identifier of the replaced node with the replacing node so that, when the identifier of the replacing node is requested, that association is used to provide the identifier of the replaced node.

[c10] 10. The method of claim 1 wherein each node has a reference count that includes a count of the snapshots through which the node is accessible.

[c11] 11. The method of claim 1 wherein the file system is a Unix-based file system.

[c12] 12. The method of claim 11 wherein a snapshot identifier is stored within each node.

[c13] 13. The method of claim 11 wherein a snapshot identifier is stored as an attribute of each node.

[c14] 14. The method of claim 1 wherein a virtual identifier is stored within a node.

[c15] 15. The method of claim 1 wherein a virtual identifier is stored as an attribute of a node.

[c16] 16. The method of claim 1 wherein when a file is modified, the new node associated with that file is set to reference the data for the modified file, rather than the data for the unmodified file.

[c17] 17. The method of claim 1 wherein when a block of a file is modified, the new node associated with that file is set to reference a block that contains the modified block, rather than the block that contains the unmodified data.

[c18] 18. The method of claim 17 including reference counting each snapshot that refers to a block so that the block can be removed when there are no more references to the block.

[c19] 19. The method of claim 18 including when the reference counting is performed using a table external to the block.

[c20] 20. The method of claim 19 wherein the table includes for each block a bit for each snapshot that indicates whether the block is referenced by the snapshot.